DuncanTech

A SPECTRUM OF SOLUTIONS

DT1100
High Resolution Digital Camera

DuncanTech
11824 Kemper Rd.
Auburn, CA 95603
Phone: (530)-888-6565
Fax: (530)-888-6579
Web: www.duncantech.com
Email: info@duncantech.com

Document Number: 9000-0004-04

DT1100 Camera User's Manual Document Number: 9000-0004-02

Reaching DuncanTech

Thank you for your purchase of a DuncanTech product. It is our priority to see that you have the highest quality product possible and the information and support necessary to get the optimum return on your investment. For further information not included in this manual, or for information on DuncanTech's other imaging products, please call:

DuncanTech
11824 Kemper Rd.
Auburn, CA 95603
Phone: (530)-888-6565
Fax: (530)-888-6579
Web: www.duncantech.com
Email: info@duncantech.com

Table of Contents

Introduction to the DT1100 Cameras

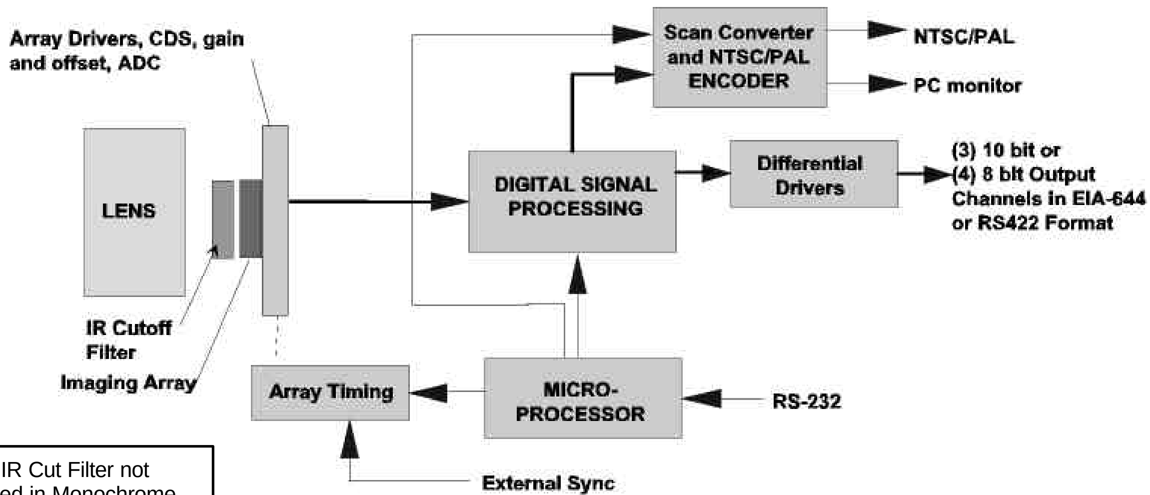
DuncanTech’s DT1100 camera is a high resolution, digital, progressive scan, area camera for monochrome and color imaging in a variety of applications. The image sensor is a charge coupled device (CCD) array sensor. Primary features of the camera are:

- Advanced optical, mechanical, and electronic design to produce high quality images without distortion.
- Progressive scan operation for clear acquisition of images of moving targets
- Digital Image Output in EIA-644 or RS-422 format.
- “Smart” camera features for advanced control and processing
- RS-232 interface for configuration and control input
- Compact, rugged, package for harsh environments
- Control of gain, offset, and exposure time for the array
- Optional analog video image output via NTSC/PAL or progressive scan
- External trigger inputs with three operating modes
- Available in two speed: 7.5 fps or 12 fps. Specify when ordering.

For detailed specifications, please see “Camera Specifications” on page 21.

Camera Operation

A functional diagram of DuncanTech's DT1100 camera is shown in Figure 1 below. The camera utilizes a single CCD sensor to acquire the image. In color configurations, the sensor has gel filters placed over the individual pixels that filter the incoming light for red, green, and blue colors. Monochrome configurations use a single, monochrome sensor.



Note: IR Cut Filter not included in Monochrome models.

Figure 1. Functional diagram - DT1100 camera

The output signal of the array is conditioned and digitized to a 10 bit digital value. Analog gain and offset of the array circuitry can be used to optimize the signal levels. The remainder of the camera electronics performs further image processing on the digital image data and outputs the data for digital transmission and/or display. For color configurations, the processing electronics converts the matrix of filtered RBG pixel values to full resolution 24 or 30 bit RGB image data in a process known as "Bayer Pattern Demultiplexing".

The camera includes an RS-232 communications interface to receive operational commands and configuration data from an external control source. An embedded microprocessor manages the communications and uses the operating parameters to configure the other camera processing units. These parameters are stored in on-board flash memory and are used to restore the camera to it's proper operating configuration at power-up.

Camera configuration and control options include the ability to set gain, offset, and exposure time independently for each channel. An internal multiplexer can be programmatically controlled to modify the mapping of image data to the digital output ports. This enables the output of any combination of color planes. Custom firmware can enable additional image processing operations such as false color look up tables, binary image plane operations, addition, subtraction, multiplication, ratioing or thresholding. External trigger inputs can be used to precisely control the start of image acquisition. Three different triggering modes are available.

Image data is output as digital pixel values at the digital output connector on the rear of the camera. Up to 32 pixels of data can be output in parallel. This output data can be programmatically configured for either 8-bit or 10-bit resolution. When 8-bit resolution is selected, the lower two bits of data are dropped. When configured for 8-bit operation, the camera can output up to four “sets” or “taps” of image data for a total of 32 bits. In 10-bit mode, the camera can output up to three “sets” or “taps” of data for a total of 30 bits. The on-board multiplexer controls which data appears at each tap. This can be any combination of processed or unprocessed image data. Digital data is output in either EIA-644 or RS-422 differential format.

An optional analog video output module adds the capability to convert the digital image data to a standard analog video format which can be output in addition to the digital data. The analog video output mode can be selected with a camera control command via the RS-232 port. Options for output format include NTSC or PAL interlaced video or progressive scan RGB at 640x480, 800x600, 1024x768, or 1280x1024.

Physical Characteristics

The DT1100 camera is housed in a compact, rugged case. Physical dimensions are shown in Figure 2 below. The maximum dimension is 89 x 97 x 107 mm without lens and cable.

NOTE: All dimensions are metric

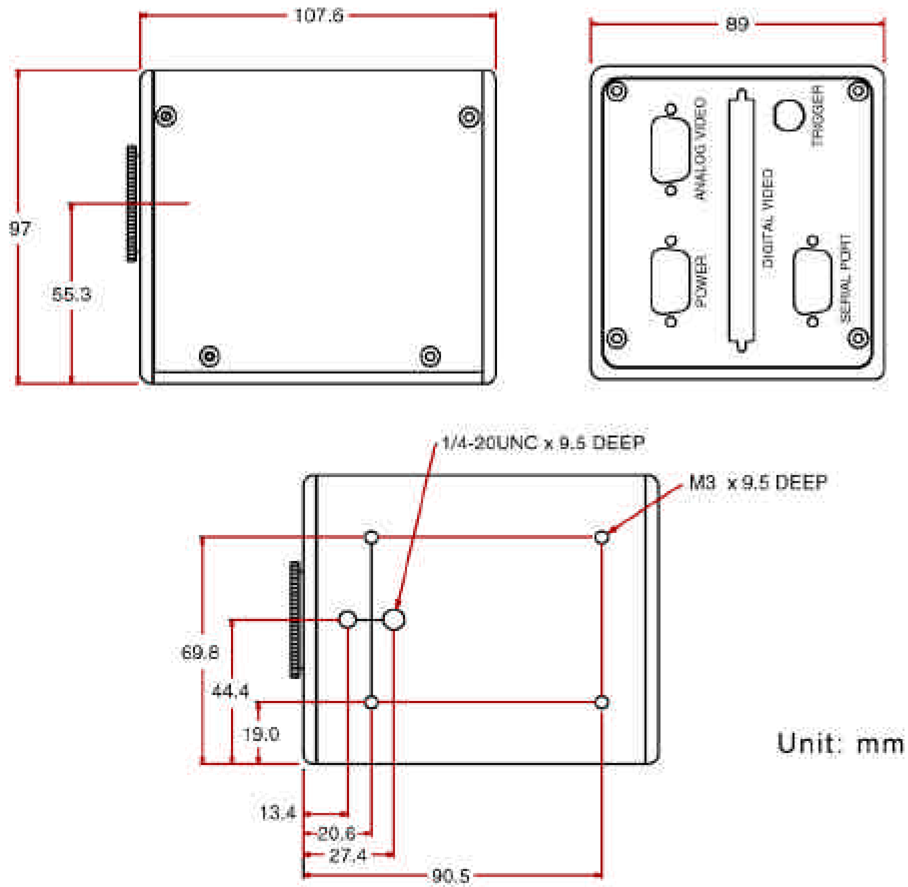


Figure 2. Camera Dimensions

Lens adapter:

The DT1100 comes with a C-Mount lens adapter. It should be noted that most standard C-Mount lenses are not designed for use with high resolution arrays and therefore do not have the small optical spot size necessary to achieve a high quality image with these arrays. For optimal image quality, you should utilize high quality, high resolution lenses.

Mounting:

For optimal stability and best heat sinking the camera should be mounted using the four threaded holes in the camera base plate (see Figure 2). The base plate is a heat sink for the camera electronics. For best performance, mount the camera to a material that provides good thermal contact and heat sinking capability.

For convenience the cameras are configured with a standard ¼-20UNC tripod mount. This is the least stable mount and is recommended only for temporary placement.

Weight without lens: .98 kg
Operating Temperature: 0-65 C
Power Supply: 12VDC, 8 Watts

Camera Electrical Interface

Overview

All electrical connectors are on the camera rear plate as shown in Figure 3. An overview of the connectors and their function is presented below followed by detailed information for each connector.

Digital Video - This connector provides access to the digital pixel data and synchronization signals from the camera. The specification for the Digital Video connector depends on the frame grabber to be used with the camera. A different output connector is provided for each supported frame grabber in order to facilitate the use of standard cables.

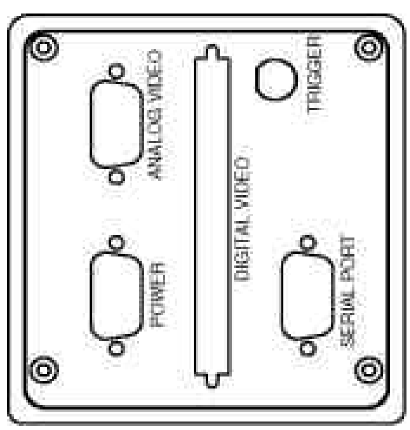


Figure 3. Camera Rear Panel

Serial Port - The RS-232 interface is provided via a standard DB-9 type connector. This provides a communications interface to send and receive configuration and control parameters.

Trigger - The external trigger input initiates the acquisition and transfer of a single frame of data. Several triggering modes are available and are configured via the RS-232 control interface.

Power - The power connector consist of a standard, DB-9 type connector. Use the power supply module provided with your camera.

Analog Video - This connector is used only in those systems that are configured with the optional analog video output board. Video output is provided on a standard DB15 connector. When the progressive scan RGB modes are selected, (640x480, 800x600, 1024x768, and 1280x1024), a standard multisync type monitor can be plugged directly into this DB15 connector. In order to drive NTSC or PAL monitors, a special DB15 to coax converter cable is included with the camera.

Digital Video Data Output

The DT1100 camera outputs up to 32 bits of parallel pixel data along with control signals for synchronization. This output can be configured as four, 8-bit parallel data channels (or “taps”) or three, 10-bit parallel data channels (or “taps”). For monochrome cameras, you can choose to use only one output, or to route the image data to more than one output tap. Output configuration is controlled via the RS-232 command interface.

The control signals PIXCLK, LVAL, and FVAL are used to clock the image data into the frame grabber. LVAL and FVAL are positive true and are coincident with the falling edge of PIXCLK. The pixel data may be latched by the rising edge of PIXCLK. This conforms to the Monochrome Digital Interface Specification AIA A15.08/3.

Control Signals

PIXCLK: Pixel clock output.
This signal is used to synchronously clock the digital video data and control signals.

LVAL: Line valid.
Asserted when a valid video line of data is being transferred.

FVAL: Frame valid.
Asserted when a valid video frame of data is being transferred.

The control signals are characterized by the following parameters and exhibit the behavior shown in the timing diagram below.

	DT1100 (7.5 fps)	DT1100 (12 fps)
Pixel Clock Rate	14.318 Mhz	22.6 Mhz
Horizontal Total Count	1790 pixels	1790 pixels
Horizontal Active Count	1392 pixels	1392 pixels
Horizontal Blank Count	398 pixels	398 pixels
Vertical Total Count	1054 lines	1054 lines
Vertical Active Count	1040 lines	1040 lines
Vertical Blank Count	14 lines	14 lines

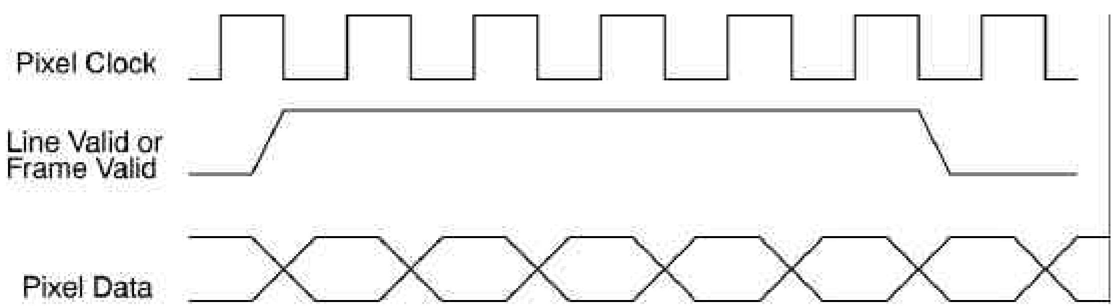


Figure 4. Timing Diagram for Digital Video Output

Digital Video Connector: Framegrabber Options
The pin assignments for various Digital Video Connectors follow.

National Instruments PCI-1424 Framegrabber

Connector: AMP786577-9 100 pin D-type subminiature

Digital Video Connector Pinout for
National Instruments PCI-1424

Pin	Signal		Pin	Signal
1	Out10+		51	Out26+
2	Out10-		52	Out26-
3	Out11+		53	Out27+
4	Out11-		54	Out27-
5	Out12+		55	Out28+
6	Out12-		56	Out28-
7	Out13+		57	Out29+
8	Out13-		58	Out29-
9	Out14+		59	Out30+
10	Out14-		60	Out30-
11	Out15+		61	Out31+
12	Out15-		62	Out31-
13	Out16+		63	Out32+
14	Out16-		64	Out32-
15	Out17+		65	Out33+
16	Out17-		66	Out33-
17	Out18+		67	Out34+
18	Out18-		68	Out34-
19	Out19+		69	Out35+
20	Out19-		70	Out35-
21	Out20+		71	Out36+
22	Out20-		72	Out36-
23	Out21+		73	Out37+
24	Out21-		74	Out37-
25	Out22+		75	Out38+
26	Out22-		76	Out38-
27	Out23+		77	Out39+
28	Out23-		78	Out39-
29	Out24+		79	Out40+
30	Out24-		80	Out40-
31	Out25+		81	Out41+
32	Out25-		82	Out41-
33	Trig1+		83	
34	Trig1-		84	
35			85	
36			86	
37			87	
38			88	
39			89	
40			90	
41	Fval+		91	
42	Fval-		92	
43	Lval+		93	RS232out
44	Lval-		94	RS232in
45	Ctrl+		95	
46	Ctrl-		96	
47			97	
48			98	
49	Pixclk+		99	Ground
50	Pixclk-		100	Ground

Imaging Technologies PC-DIG Framegrabber
Connector: AMP175925-9 100 pin D-type subminiature

Digital Video Connector Pinout for
Imaging Technology PC-DIG

Pin	Signal		Pin	Signal
1	Out10+		51	Out26+
2	Out10-		52	Out26-
3	Out11+		53	Out27+
4	Out11-		54	Out27-
5	Out12+		55	Out28+
6	Out12-		56	Out28-
7	Out13+		57	Out29+
8	Out13-		58	Out29-
9	Out14+		59	Out30+
10	Out14-		60	Out30-
11	Out15+		61	Out31+
12	Out15-		62	Out31-
13	Out16+		63	Out32+
14	Out16-		64	Out32-
15	Out17+		65	Out33+
16	Out17-		66	Out33-
17	Out18+		67	Out34+
18	Out18-		68	Out34-
19	Out19+		69	Out35+
20	Out19-		70	Out35-
21	Out20+		71	Out36+
22	Out20-		72	Out36-
23	Out21+		73	Out37+
24	Out21-		74	Out37-
25	Out22+		75	Out38+
26	Out22-		76	Out38-
27	Out23+		77	Out39+
28	Out23-		78	Out39-
29	Out24+		79	Out40+
30	Out24-		80	Out40-
31	Out25+		81	Out41+
32	Out25-		82	Out41-
33	Lval+		83	
34	Lval-		84	
35	Fval+		85	
36	Fval-		86	
37	Ground		87	
38	Ground		88	
39	Pixclk+		89	
40	Pixclk-		90	
41			91	
42			92	
43			93	
44			94	
45			95	
46			96	
47	Ctrl+		97	
48	Ctrl-		98	
49			99	
50			100	

Matrox Corona
Connector: AMP175925-0 80 pin D-type subminiature

Digital Video Connector Pinout for
Matrox Corona

Pin	Signal		Pin	Signal
1	Out10+		41	Out11+
2	Out10-		42	Out11-
3	Out12+		43	Out13+
4	Out12-		44	Out13-
5	Out14+		45	Out15+
6	Out14-		46	Out15-
7	Out16+		47	Out17+
8	Out16-		48	Out17-
9	Out18+		49	Out19+
10	Out18-		50	Out19-
11	Out20+		51	Out21+
12	Out20-		52	Out21-
13	Out22+		53	Out23+
14	Out22-		54	Out23-
15	Out24+		55	Out25+
16	Out24-		56	Out25-
17	Out26+		57	Out27+
18	Out26-		58	Out27-
19	Out28+		59	Out29+
20	Out28-		60	Out29-
21	Out30+		61	Out31+
22	Out30-		62	Out31-
23	Out32+		63	Out33+
24	Out32-		64	Out33-
25	Ground		65	Ground
26			66	
27			67	Lval+
28			68	Lval-
29			69	Fval+
30			70	Fval-
31	Trig1+		71	
32	Trig1-		72	
33			73	Ctrl+
34			74	Ctrl-
35			75	
36			76	
37			77	
38			78	
39			79	Pixclk+
40			80	Pixclk-

Serial Port Communication Interface

The RS-232 interface to the camera is provided via a standard, DB-9 type connector on the rear panel with the following connections. The data character format is 8N1 (8 data bits + no parity + 1 stop bit). Baud rate is 9600 bps. No handshaking signals are supported. For detailed information on the command protocol, see "Camera Control and Configuration via RS-232 Communications".

RS-232 Connector Pin Assignments

Pin	Connection	Notes
2	Transmit	Host PC output
3	Receive	Host PC input
5	Ground	

Trigger Input

The external trigger signal initiates the acquisition and transfer of a single frame of data in one of several possible ways. The polarity of the External Trigger signal is user programmable. The external trigger signal input may be derived from one of two sources: 1) the Trigger BNC connector on the rear panel or 2) the trigger signal pins on the Digital Video Connector. The source of the trigger input is selected via an RS-232 command. The optically coupled, rear panel BNC input requires a trigger voltage from 4 to 10 volts in amplitude and capable of sourcing at least 10 mA.

Trigger Modes

Image acquisition occurs in four different modes. Three of these modes require an external trigger signal to initiate a new acquisition. These triggered modes provide different methods of controlling the start of image acquisition and the duration of the exposure time. The triggering mode is selected with via an RS-232 command. The triggering modes are described in detail below.

<u>Free Run Mode</u> (Internal Sync)

This mode requires no external control signals and provides high frame rates by overlapping the readout time with the exposure time. An internally generated, fixed frequency trigger signal initiates the readout of the current frame and starts the exposure time for the next frame. The frame rate is controlled internally. Exposure time is programmable.

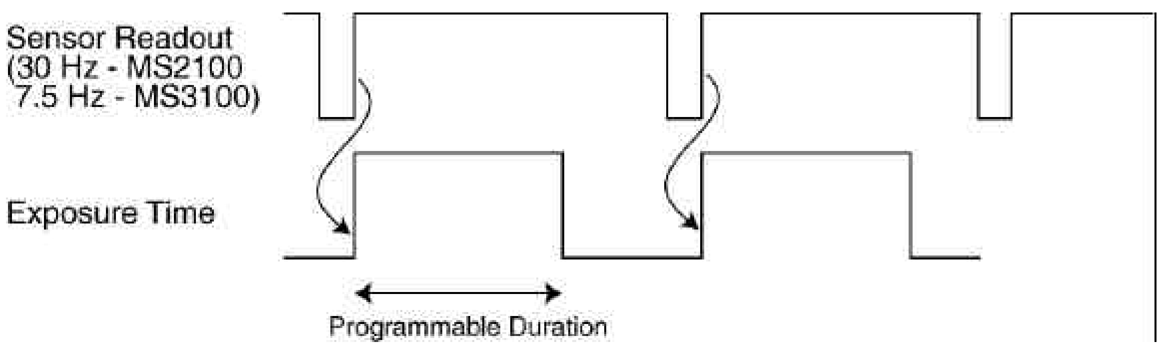


Figure 5. Free Run Mode - No External Trigger

<u>Edge Controlled</u> (External Trig)
This mode provides high frame rates by overlapping the readout time with the exposure time. The active edge of EXT TRIG initiates the readout of the last frame of data and starts the exposure time for the next frame. The exposure time is defined by the time between two successive leading edges of the trigger signal. The minimum time between trigger pulses must be at least one frame readout period.

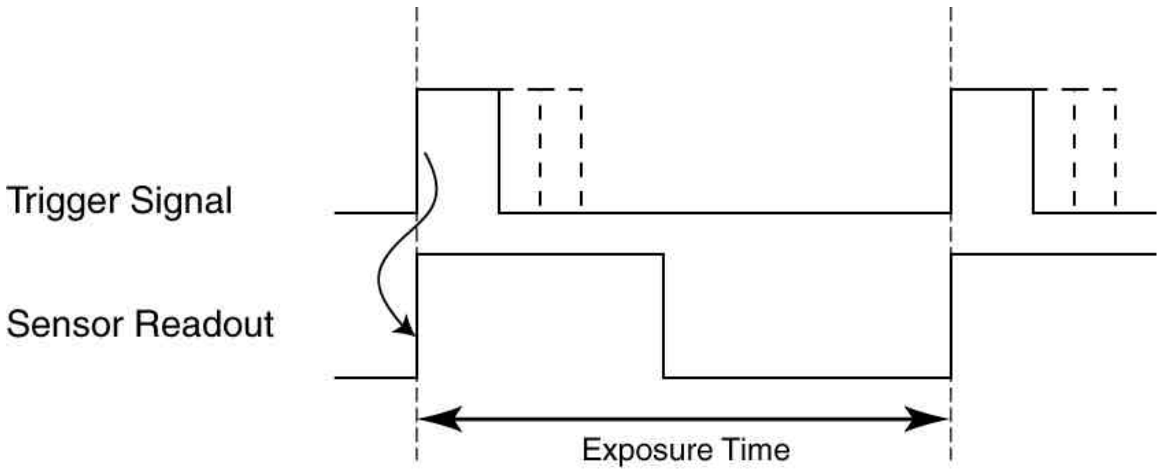


Figure 6. Edge Controlled Trigger Mode

<u>Integrate and Dump</u> (External Trig, programmable)
In this mode the active edge of EXT TRIG initiates the start of a programmable exposure time. At the end of the exposure time the readout takes place. After the readout, the system is ready for another EXT TRIG signal. The exposure time programmed via RS-232 commands and can be set with CameraLink, DuncanTech's camera control program.

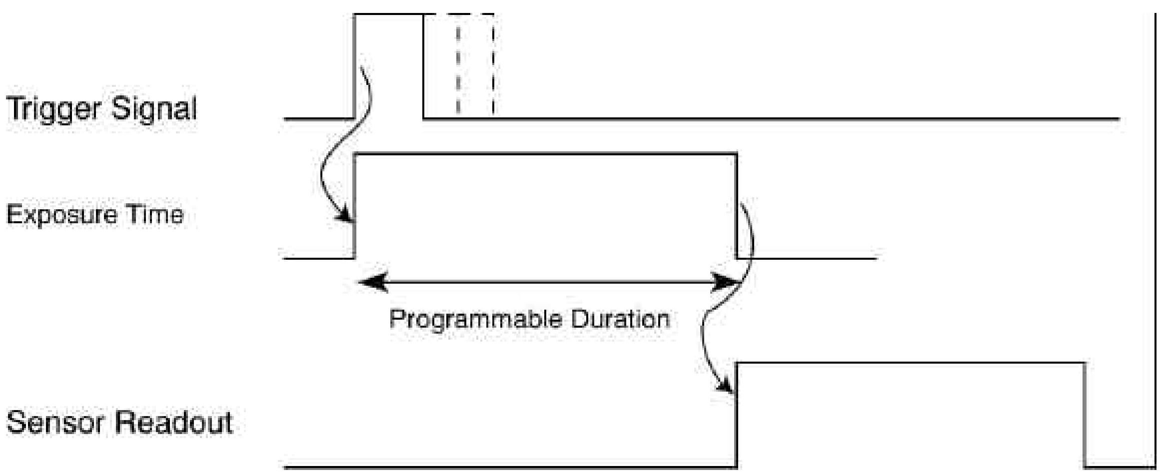


Figure 7. Integrate & Dump - Edge Controlled Trigger Mode

<u>Integrate and Dump</u> (External Trig, level controlled)
In this mode both edges of EXT TRIG are active. The leading edge initiates the start of the exposure time and the falling edge defines the end of the exposure time. The falling edge also initiates the readout period. The minimum time between two successive leading edges of the trigger signal is the exposure time plus one frame readout period..

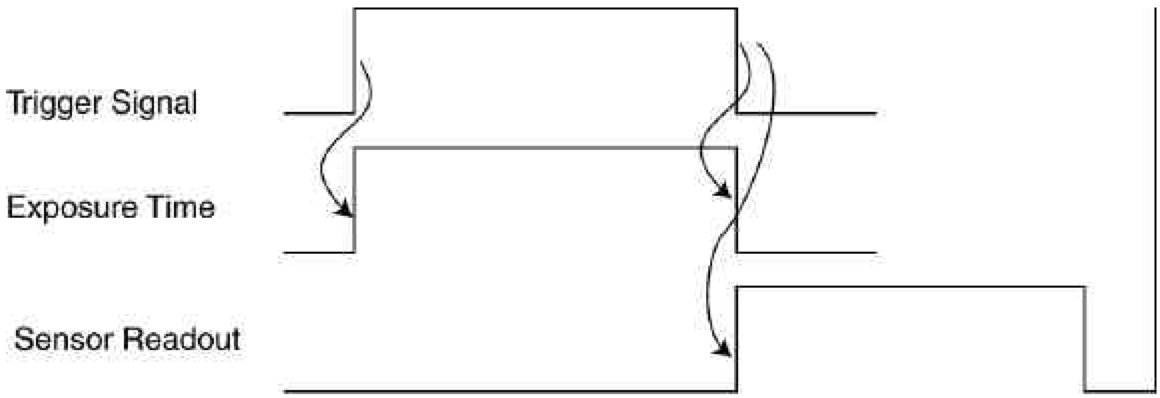


Figure 8. Integrate & Dump - Level Controlled Trigger Mode

Electrical Power Requirements

The DT1100 series camera has built-in power conditioning. The camera requires 12Volts +/- 5% at 1 amps .

Maximum power dissipation for the available camera models is shown below.

Maximum Power Dissipation

	Basic Camera	Full Featured*
DT1100	7 W	12 W

*Includes signal processor, analog video output, and flash memory.

The power connector consists of a standard, DB-9 type connector on the rear panel of the camera with the following connections.

Power Connector Pinout

Pin	Connection
1	Ground
2	+12V

Video Output

For those cameras purchased with the analog video output options, the analog video is available on a DB15 connector located on the camera rear panel. Video output is provided in NTSC or PAL formats (Composite or S-Video) as well as non-interlaced video for multisync (PC type) monitors. The format of the video output signal is selected via the RS232 interface. A standard DB15 to coax cable can be used to interface with NTSC/PAL monitors or to provide access to the RED, GREEN, BLUE and Sync outputs for multisync monitors or analog frame grabbing operations.

Analog Video Connector Pinout

PIN	Progressive Scan Output	NTSC/PAL Output
1	Red	S-Video (C)
2	Green	Composite Video
3	Blue	S-Video (Y)
4	N/C	
5	N/C	
6	Red Ground	Video Gnd
7	Green Ground	Video Gnd
8	Blue Ground	Video Gnd
9	N/C	
10	Ground	
11	N/C	
12	N/C	
13	Horiz. Sync	
14	Vert. Sync	
15	N/C	

DB15 to Coax Cable

Coax	NTSC/PAL Output	Interlaced RGB Output
Red	S-Video (C)	Red
Green	Composite Video	Green
Blue	S-Video (Y)	Blue
White or Gray		Composite Sync

Supported Video Modes

The table below lists the video signal formats that can be output from the Analog Video connector. The output mode is selected via a camera control command. An appropriate monitor type that can support the selected mode must be used to view the resulting image.

Mode	Resolution	Line Rate	Frame Rate	Pixel Clock Rate	
NTSC	640 X 504	15.7 KHz	60 Hz Intl	12.27 MHz	(interlaced)
PAL	768 X 600	15.6 KHz	50 Hz Intl	14.75 MHz	(interlaced)
VGA (640 x 480)	640 X 480	31.5 KHz	60 Hz	25.175 MHz	
800 x 600	800 X 600	37.9 KHz	60 Hz	40 MHz	
1024 x 768	1024 X 768	48.4 KH z	60 Hz	65 MHz	*future
1280 x 1024	1280 X 1024	64.0 KHz	60 Hz	108 MHz	

Not all video modes are valid on all camera models. The table below lists camera models cross-referenced to video modes.

Valid Video Modes per Camera Model

Video Mode	DT1100 (1392x1040)
NTSC	Yes
PAL	Yes
640x480 *	Yes
800x600 *	Yes
1024x768 *	Yes
1280x1024 *	Yes

Camera Specifications

	DT1100
Imaging Device	½ in Interline Transfer CCD
Resolution	1392(H) x 1040(V)
Pixel Size	4.65 x 4.65 micron
Pixel Clock Rate	14.318 MHz max
Sensing Area	7.6 x 6.2 mm
Frame Rate	7.5 frames per second
Digital Output	8 bits x 4 taps or 10 bits x 3 taps (32 bits max) EIA644 or RS422
Data Transfer Rate	14.318 MHz (7.5 fps) or 22.6 MHz (12 fps)
Digital Control Signals	Pixclk, Fval, Lval, Ext Trigger
Signal/Noise	60 dB
Lens Mount	C Mount
Electronic Shutter	1/8000 - 1/7.5 sec Independent control per channel
Gain Selection	-4 - 32 dB Independent control per channel
Offset Selection	0-127 counts Independent control per channel
External Trigger Input	BNC or Digital Video Connector
Control Input	RS-232 port
Operating Temp	0-65° C
Operating Voltage	12 volts
Power Consumption	8 Watts
Weight	.98

Camera Control and Configuration via RS-232 Communications

RS-232 Command Set

Communication between the host and the camera takes place by way of the transmission of message packets from one to the other. Communication is always initiated by the host in the form of a host message packet (described below). The camera responds with an echo message packet (described below) which may or may not contain message bytes. Commands that perform functions (such as setting parameters) are echoed back to the host <u>after</u> the function has been performed, with no message bytes. A status flag indicates if the action was successful or not. Commands from the host that expect data in return (like getting gain or offset values) are echoed with the requested data in the form of message bytes along with a status flag which indicates if the action was successful or not.

Note: The camera requires that a command sequence be executed in a handshaking fashion. When the host has sent a command and is waiting for the echoed response from the camera, no additional commands may be sent to the camera. New commands may only be sent to the camera when the previous command has been completed and the status echo received. Violating this rule may result in unpredictable results.

Host Message Format

The format for all messages transmitted to the camera will be:

STX <"size of message" LSB> <"size of message" MSB> <command byte> <message bytes> <checksum byte>

where:

STX =>	ASCII Start Transmission character ($02).
"size of message" LSB =>	Least-significant byte of 16-bit *size of message* field. (Note that the size value does ***not*** include the **STX** byte, the "*size of message"* bytes, or the checksum byte.)
"size of message" MSB =>	Most-significant byte of 16-bit "*size of message"* field.
command byte =>	Unique byte for each host command
message bytes =>	Zero or more message/data bytes. (Exact number determined by the parameters of the command.)
checksum =>	8 bit, two's complement of sum of message bytes (does not include STX or "*size of message"* bytes)

Checksum calculation: In order to calculate the check sum for any given command, accumulate the 8-bit sum off all bytes that constitute the command and it's message bytes. Do NOT include the STX and "size of message" bytes in this sum. Having accumulated this sum, take the twos compliment of the sum. This will be the command checksum value. In C, the twos compliment of the sum <*sumval*> can be calculated as:

<Twos comp val> = -<sumval>;

Echo Message Format

Once a command has been received at the camera, it will be processed and the command will be echoed back to the host for verification. The format for all echoed messages transmitted from the camera is:

STX <"size of message" LSB> <"size of message" MSB> <command byte> <message bytes> <status byte> <checksum byte>

where:

STX =>	ASCII Start Transmission character ($02).
"size of message" LSB =>	Least-significant byte of 16-bit *size of message* field. (Note that the size value does ***not*** include the **STX** byte, the *size of message* bytes or the checksum byte.)
"size of message" MSB =>	Most-significant byte of 16-bit *size of message* field giving the number of bytes to follow in message field.
command byte	Unique byte for each host command
message bytes	Zero or more message/data bytes.
status byte	Indicates success or failure of the disposition of the command
checksum	8 bit, two's complement of sum of message bytes (does not include STX or size bytes but does include the status byte)

Allowable values for the status bytes include:

CommandComplete	**0x00**	Command executed without error.
CommandFailure	**0x01**	Command execution failed.
ChecksumFailure	**0x02**	Checksum calculation failed
UnrecognizedCommand	**0x03**	Command was not recognized

Camera Command Set

The following lists each command that is recognized by the camera, it's parameters, structure, and expected echo.

Definition of Channel Number

Many of the commands that follow will refer to "Channel Number" as a parameter. The meaning and value of this parameter will vary depending on what model of camera you are working with.

The camera-imaging engine includes three channels for image data. In 3-CCD cameras, there is one channel per CCD sensor. In 1-CCD cameras, there is only sensor which is controlled by the electronics for "Channel 3". In remote head cameras, the channel values used in the commands correspond to Head 1, 2, and 3 respectively. Assignment of head numbers is determined by which connector on the front panel a given head is plugged into.

The correlation between the Channel No parameter for the following commands and the various camera models is detailed in the table below.

Table 1. Definition of Channel No Parameter

Camera Configuration	Channel 1	Channel 2	Channel 3
MS2100-RGB MS2150-RGB MS3100-RGB	Green Sensor	Red Sensor	Blue Sensor
MS2100-CIR MS2150-CIR MS3100-CIR	Red Sensor	IR Sensor	Green Sensor
MS2100-RGB/CIR MS2150-RGB/CIR MS3100-RGB/CIR	Red Sensor	IR Sensor	Blue/Green Sensor
MS2200-RGB	Green Sensor	Red Sensor	Blue Sensor
MS2200-CIR	Red Sensor	IR Sensor	Green Sensor
DT1100-RGB DT1100-Mono	Not used	Not used	RGB or Mono Sensor
DT1200	Not used	Not used	Linear Sensor
RH1100	Head 1 Sensor	Head 2 Sensor	Head 3 Sensor

SetChannelGain(ChannelNumber, Gain)

Sets the specified channel to the specified gain value where:

ChannelNumber = 1, 2, or 3
Gain is a 16-bit value calculated as follows:

For MS2100 and MS2150
Gain = 0 - 384 where the resulting gain value in dB is calculated as:
gain = (.094)*DigitalNumber-4 dB

For all other DuncanTech Cameras
Gain = 95 – 1023 counts where the resulting gain value is calculated as:
gain =. (0366)*(DigitalNumber-95)+2.0 dB

Message Byte	Contents
0	$02 - STX
1	$04 - LSB size
2	$00 - MSB size
3	$02 - command byte
4	channel number
5	gain - LSB
6	gain - MSB
7	$?? - checksum

Echo:

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$02 - command byte
4	status
5	$?? - checksum

GetChannelGain(ChannelNumber)

Requests the camera to return the present gain setting for the specified channel. Returns message bytes and status.

ChannelNumber = 1, 2, or 3

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$03 - command byte
4	channel number
5	$?? - checksum

Echo:

Message Byte	Contents
0	$02 - STX
1	$05 - LSB size
2	$00 - MSB size
3	$03 - command byte
4 5	channel number gain (LSB)
6	gain (MSB)
7	$?? - status
8	$?? - checksum

SetChannelOffset(ChannelNumber, Offset)

Adds the specified offset to the specified channel. The offset value is in an 8 bit, straight binary format. [Note: DuncanTech's Correrlated Double Sampling circuitry automatically removes any offset at the beginning of every line. Due to this advanced technology, it has been found that this particular command is not needed because image signal does not have an offset. However, the offset command is accessible programmatically and may have utility for specific applications. The comannd continues to be included her for completeness.

ChannelNumber = 1, 2, or 3
For Cameras with AD9841
Offset = 0 – 127
For Cameras with AD9841
Offset = 0 - 63

Message Byte	Contents
0	$02 - STX
1	$03 - LSB size
2	$00 - MSB size
3	$04 - command byte
4	channel number
5	offset
6	$?? - checksum

Echo:

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$04 - command byte
4	status
5	$?? - checksum

GetChannelOffset(ChannelNumber)

Requests the camera to return the present offset setting for the specified channel. The offset value is in an 8 bit, straight binary format.

ChannelNumber = 1, 2, or 3

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$05 - command byte
4	channel number
5	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$04 - LSB size
2	$00 - MSB size
3	$05 - command byte
4 5	channel number offset
6	$?? - status
7	$?? - checksum

SetIntegrationTime(ChannelNumber, IntegrationTime)

Note: This command title adjusts the length of the time period during which the sensor gathers light for any given frame. In a conceptual sense, this is often thought of and referred to as "exposure" control. However, this terminology can be confusing. The specific parameter being adjusted is the integration time period for the sensor. Parameters for the command include:

ChannelNumber =	1, 2, or 3
IntegrationTime =	1 - 500 (MS2100)
	1 - 588 (MS2150)
	1 - 1046 (MS3100)
	1 - 1046 (DT1100-4)
	1 - 1071 (MS2200 – 1024x1)
	1 – 1071 (DT1200 – 1024x1)
	1 - 2098 (DT1200 – 2048x1)
	1 - 1046 (RH1100)
	1 - 1046 (RH1200 – 1024x1)

Note: The integration time parameter specifies the number of line periods (i.e. the time required to read one line of the image) that should elapse for the integration period. Conversion of this unitless value to an integration time value in seconds is therefore a function of line length (the # of pixels in a row and the pixel clock rate). Values for the various camera models are as follows:

Model	PixClk MHz	Fps	Max Cnt	Min (msec)	Max (msec)	Incr (msec)
MS2100	12	30	500	.065	32.5	.065
MS2150	14.07	25	588	.067	39.5	.067
MS3100 (7.5 fps)	14.318	7.59	1046	.125	130.75	.125
MS3100 (10 fps)	18.87	10	1046	.095	99	.095
DT1100-4 (7.5 fps)	14.318	7.59	1046	.125	130.75	.125
DT1100-4 (12 fps)	22.6	12	1046	.079	83	.079
RH1100-4 (7.5 fps)	14.318	7.59	1046	.125	130.75	.125
RH1100-4 (10 fps)						
RH1100-4 (12 fps)	22.6	12	1046	.079	83	.079
MS2200 (1024x1)				1	1071	1
DT1200				1	1071	1
RH1200 (1024x1)				1	1071	1
RH2200 (1024x1)				1	1071	1

Message Byte	Contents
0	`$02 - STX`
1	`$04 - LSB size`
2	`$00 - MSB size`
3	`$14 - command byte`
4	`channel number`
5	`Exposure Time - LSB`
6	`Exposure Time - MSB`
7	`$?? - checksum`

Echo:

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$14 - command byte
4	status
5	$?? - checksum

GetIntegrationTime(ChannelNumber)

Requests the camera to return the present integration time setting for the specified channel. The returned value represents the number of scan lines that go to make up the integration time. To convert this count to msec, multiply the returned value by the Incr value in the table above.

ChannelNumber = 1, 2, or 3

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$15 - command byte
4	channel number
5	$?? - checksum

Echo:

Message Byte	Contents
0	$02 - STX
1	$05 - LSB size
2	$00 - MSB size
3	$15 - command byte
4 5	channel number Exposure Time (LSB)
6	Exposure Time (MSB)
7	$?? - status
8	$?? - checksum

SetTriggerMode()

Sets the camera to one of the supported trigger modes.

Message Byte Structure:

SetTriggerMode()

LSB Bits 0,1,2 = Area Camera Modes
- 0 = Video mode
- 1 = Edge mode
- 2 = Int & Dump, level mode
- 3 = Int & Dump, programmable, ganged
- 4 = Int & Dump, programmable, individual
- 5-7 = Unused

LSB Bits 0,1,2 = Line Camera Modes
- 0 = Frame Mode, Free Running
- 1 = Frame Mode, Triggered
- 2 = Line Mode, Free Running
- 3 = Line Mode, Edge Triggered
- 4 = Line Mode, Int & Dump, Level Controlled
- 5 = Line Mode, Int & Dump, Programmable
- 6-7 = Unused

LSB Bit 3 = Trigger Source
- 0 = BNC
- 1 = Frame Grabber

LSB Bit 4 = Trigger Polarity
- 1 = Positive Edge or Level
- 0 = Negative Edge or Level

Message Byte	Contents
0	$02 - STX
1	$03 - LSB size
2	$00 - MSB size
3	$16 - command byte
4 5	Mode - LSB Mode - MSB
6	$?? - checksum

Echo:

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$16 - command byte
4	status
5	$?? - checksum

GetTriggerMode()

Requests the camera to return the present trigger mode setting.

Message Byte	Contents
0	$02 - STX
1	$01 - LSB size
2	$00 - MSB size
3	$17 - command byte
4	$?? - checksum

Echo:

Message Byte	Contents
0	$02 - STX
1	$04 - LSB size
2	$00 - MSB size
3	$17 - command byte
4	trigger mode (LSB)
5	trigger mode (MSB)
6	$?? - status
7	$?? - checksum

SetOutputMux(Three Byte Value)

Sends a three-byte message to the camera specifying the camera multiplexing configuration. This determines how the available data from the camera is mapped to the output ports or taps. The correspondence between Ports and display color plane is a function of the receiving frame grabber or host circuitry. Typically, the analog video output is configured such that, Port 0 corresponds to red, Port 1 corresponds to blue, and Port 2 corresponds to green. However, this mapping can be changed with the SetVideoMux command. Port four has no meaning for the analog video output.

This command determines which camera image will be directed from to any given digital output port in the system.

This command also includes the ability to set a digital multiplier for each channel. This causes the digital pixel value for all the pixels of a given channel to be multiplied by either one, two, or four (performing a left shift).

Message Byte Structure:

Byte0 Bits;
- 0,1,2 = Port0 Array Select
  - (0=Array1
  - 1=Array2
  - 2=Array3
  - 3=Processed Red
  - 4 = Processed Green
  - 5 =Processed Blue
  - 6=Processed Mono
  - 7=Off)
- 3,4,5 = Port1 Array Select (see Port0 Array Select)
- 6,7 = Unused

Byte1 Bits;
- 0,1,2 = Port2 Array Select (see Port0 Array Select)
- 3,4,5 = Port3 Array Select (see Port0 Array Select)
- 6 = Unused
- 7 = Data Resolution
  - (0 = 8 bits
  - 1 = 10 bits)

Byte2 Bits;
- 0,1 = Array1 Multiplier (0=X1, 1=X2, 2=X4)
- 2,3 = Array2 Multiplier
- 4,5 = Array3 Multiplier
- 6,7 = Unused

Message Byte	Contents
0	$02 - STX
1	$04 - LSB size
2	$00 - MSB size
3	$1A - command byte
4	Value - Byte0
5	Value - Byte1
6	Value - Byte2
7	$?? - checksum

Echo:

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$1A - command byte
4	status
5	$?? - checksum

GetOutputMux()

Requests the camera to return the present output mux configuration.

Message Byte	Contents
0	$02 - STX
1	$01 - LSB size
2	$00 - MSB size
3	$1B - command byte
4	$?? - checksum

Echo:

Message Byte	Contents
0	$02 - STX
1	$05 - LSB size
2	$00 - MSB size
3	$1B - command byte
4	Value - Byte0
5	Value - Byte1
6	Value - Byte2
7	$?? - status
8	$?? - checksum

SetVideoMode(Value)

This command provides a means to configure the optional analog video output. In cameras that were purchased without the DirectView video option, this command will return an Unknown Command status from the camera. This command can also be used to cause the camera to output a color bar pattern for system test and setup.

The value passed determines the format that will be used for the video output signal. Not all modes are valid for all camera models. The table below shows the available video formats, which ones are valid for each camera model, and the value that should be passed to set that mode.

Video Mode	**MS2100 (656x494)**	**MS2150 (780x582)**	**MS3100 (1392x1040)**	**RH1100 (1392x1040)**	**MS2200 (1024 Line)**
NTSC	Yes	Yes	Yes	Yes	Yes
PAL	Yes	Yes	Yes	Yes	Yes
Interlaced RGB	Yes	NA	NA	NA	NA
640x480 *	Yes	Yes	Yes	Yes	Yes
800x600 *	NA	Yes	NA	NA	NA
1024x768 *	NA	NA	NA	NA	Yes
1280x1024 *	NA	NA	Yes	Yes	NA

* - Progressive Scan RGB

Message Byte Structure:

Value = 2 bytes

LSB, Bits 0,1,2 = Mode Select
(See table above)

LSB, Bit 3 = Gamma
(0 = no NTSC gamma correction
1 = NTSC gamma correction enabled)

LSB, Bit 4,5 = Output
0 = normal video output
1 = color bar pattern output
2-3 = unused

LSB, Bit 6 = unused

LSB, Bit 7 = reserved, must be zero

MSB, Bits 0,1,2 = PAL mode
0 = B
1 = D
2 = G
3 = H
4 = I
5 = M
6 = N

Message Byte	Contents
0	$02 - STX
1	$03 - LSB size
2	$00 - MSB size
3	$1C - command byte
4 5	Value - LSB Value - MSB
6	$?? - checksum

Echo:

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$1C - command byte
4	status
5	$?? - checksum

GetVideoMode()

Requests the camera to return the current video mode configuration.

Message Byte	Contents
0	$02 - STX
1	$01 - LSB size
2	$00 - MSB size
3	$1D - command byte
4	$?? - checksum

Echo:

Message Byte	Contents
0	$02 - STX
1	$04 - LSB size
2	$00 - MSB size
3	$1D - command byte
4	Value - LSB
5	Value - MSB
6	$?? - status
7	$?? - checksum

CorrectOffset(ChannelNumber) – Linescan cameras only

This command is used for the flat field normalization process in lines scan cameras. It will have no effect in area scan cameras. Applies offset correction to the specified channel.

ChannelNumber = 1 - 3

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$36 - command byte
4	channel number
5	$?? - checksum

Echo:

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$36 - command byte
4	status
5	$?? - checksum

GetOffsetCorrectionResult(ChannelNumber) – Line Scan Cameras Only

Returns the average value of the pixels determined during the offset correction process for the specified channel.

ChannelNumber = 1 - 3

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$37 – command byte
4	channel number
5	$?? – checksum

Echo:

Message Byte	Contents
0	$02 – STX
1	$05 - LSB size
2	$00 - MSB size
3	$37 – command byte
4	channel number
5	Average Value - Low Byte
6	Average Value - High Byte
7	Status
8	$?? – checksum

CorrectGain(ChannelNumber)

This command is used for the flat field normalization process in lines scan cameras. It will have no effect in area scan cameras. Applies gain correction to the specified channel.

ChannelNumber = 1 - 3

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$38 - command byte
4	channel number
5	$?? - checksum

Echo:

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$38 - command byte
4	status
5	$?? - checksum

GetGainCorrectionResult(ChannelNumber)

Returns the maximum value of the pixels determined during the gain correction process for the specified channel.

ChannelNumber = 1 - 3

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$39 - command byte
4	channel number
5	$?? - checksum

Echo:

Message Byte	Contents
0	$02 - STX
1	$05 - LSB size
2	$00 - MSB size
3	$39 - command byte
4 5 6 7	channel number Maximum Value - Low Byte Maximum Value - High Byte status
8	$?? - checksum

SetPixelClockRate(Frequency)

Line Scan Cameras Only

Caution: DuncanTech area scan cameras are built for a specified pixel clock rate. Do not use this command on an area scan camera. Changing the pixel clock rate of area scan models may cause the camera to malfunction.

This command allows you to request a different pixel clock speed for a linescan camera. Anytime the pixel clock for the camera is changed, you should power down the camera and then re-start it. Changing the pixel clock without cycling the power may result in unpredictable behavior.

Frequency = in integer units of Mhz

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$0A - command byte
4	Frequency
5	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$0A - command byte
4	status
5	$?? - checksum

GetPixelClockRate()

Message Byte	Contents
0	$02 - STX
1	$01 - LSB size
2	$00 - MSB size
3	$0B - command byte
4	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$03 - LSB size
2	$00 - MSB size
3	$0B - command byte
4	Frequency
5	$?? - status
6	$?? - checksum

SetAnalogColorBalance()

This command is used with camera models that utilize a Bayer Pattern Color Filter CCD sensor. This includes the DT1100, RH1100 with color heads, and MS3100-RGB/CIR. Each color may be multiplied by a six-bit value corresponding to a scaling of -2dB to +10Db. Commmand Parameters:

Byte 0 = Red scale factor
Byte 1 = Green scale factor
Byte 2 = Blue scale factor

Message Byte	Contents
0	$02 - STX
1	$04 - LSB size
2	$00 - MSB size
3	$30 - command byte
4 5 6	Red Scale Factor Green Scale Factor Blue Scale Factor
7	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$30 - command byte
4	status
5	$?? - checksum

GetAnalogColorBalance()

This command is used with camera models that utilize a Bayer Pattern Color Filter CCD sensor. This includes the DT1100, RH1100 with color heads, and MS3100-RGB/CIR. Returns the color balance scale factors for red, green and blue as well as three, 16 bit values corresponding the average intensity of each color in the color balance measurement window.

Byte 0 = Red scale factor
Byte 1 = Green scale factor
Byte 2 = Blue scale factor
Byte 3 = Red Intensity LSB
Byte 4 = Red Intensity MSB
Byte 5 = Green Intensity LSB
Byte 6 = Green Intensity MSB
Byte 7 = Blue Intensity LSB
Byte 8 = Blue Intensity MSB

Message Byte	Contents
0	$02 - STX
1	$01 - LSB size
2	$00 - MSB size
3	$31 - command byte
4	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$0b - LSB size
2	$00 - MSB size
3	$31 - command byte
4	Red scale factor
5	Green scale factor
6	Blue scale factor
7	Red Intensity LSB
8	Red Intensity MSB
9	Green Intensity LSB
10	Green Intensity MSB
11	Blue Intensity LSB
12	Blue Intensity MSB
13	$?? - status
14	$?? - checksum

SetZoomFactor()

This command is only available in cameras with DirectView Analog Video. For cameras with DirectView, availability of this command will depend upon the rev level of the hardware.

Sets X1, X2 or X4 zoom factor for video display
Byte 0 = ZoomFactor;
1 = X1
2 = X2
4 = X4

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$32 - command byte
4	ZoomFactor
5	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$32 - command byte
4	status
5	$?? - checksum

GetZoom Factor()

Returns the video zoom factor
Byte 0 = ZoomFactor;
1 = X1
2 = X2
4 = X4

Message Byte	Contents
0	$02 - STX
1	$01 - LSB size
2	$00 - MSB size
3	$33 - command byte
4	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$03 - LSB size
2	$00 - MSB size
3	$33 - command byte
4	ZoomFactor
5	$?? - status
6	$?? - checksum

SetVideoMux()

Note: This command should be used with caution. It should only be necessary to change these settings when the color plane mapping between the digital output ports and the analog video ports do not correspond, resulting in color differences between the analog video display and the digital display. Some framegrabbers require this correction.

Sets Video Card Multiplexer Configuration

Byte 0 = MuxConfiguration

Bits;

0 - 1 = Red Output Selection

0 = Port0

1 = Port1

2 = Port2

3 = Port3

2 - 3 = Green Output Selection

4 - 5 = Blue Output Selection

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$3d - command byte
4	MuxConfiguration
5	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$3d - command byte
4	status
5	$?? - checksum

GetVideoMux()

Gets Video Card Multiplexer Configuration
Byte 0 = MuxConfiguration
Bits;
0 - 1 = Red Output Selection
0 = Port0
1 = Port1
2 = Port2
3 = Port3
2 - 3 = Green Output Selection
4 - 5 = Blue Output Selection

Message Byte	Contents
0	$02 - STX
1	$01 - LSB size
2	$00 - MSB size
3	$3e - command byte
4	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$03 - LSB size
2	$00 - MSB size
3	$3e - command byte
4	MuxConfiguration
5	$?? - status
6	$?? - checksum

SetCrosshairs()

Sets Crosshairs in digital image data Byte 0 = CrosshairControl
Bit 0 = On/Off;
0 = Off
1 = On

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$3f - command byte
4	CrosshairControl
5	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$3f - command byte
4	status
5	$?? - checksum

GetCrosshairs ()

Gets Crosshair Status
Byte 0 = CrosshairControl
Bit 0 = On/Off;
0 = Off
1 = On

Message Byte	Contents
0	$02 - STX
1	$01 - LSB size
2	$00 - MSB size
3	$40 - command byte
4	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$03 - LSB size
2	$00 - MSB size
3	$40 - command byte
4	CrosshairControl
5	$?? - status
6	$?? - checksum

SetZoomFactor()

This command is only available in cameras with DirectView Analog Video. For cameras with DirectView, availability of this command will depend upon the rev level of the hardware.

Sets X1, X2 or X4 zoom factor for video display
Byte 0 = ZoomFactor;
1 = X1
2 = X2
4 = X4

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$32 - command byte
4	ZoomFactor
5	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$32 - command byte
4	status
5	$?? - checksum

GetZoom Factor()

Returns the video zoom factor
Byte 0 = ZoomFactor;
1 = X1
2 = X2
4 = X4

Message Byte	Contents
0	$02 - STX
1	$01 - LSB size
2	$00 - MSB size
3	$33 - command byte
4	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$03 - LSB size
2	$00 - MSB size
3	$33 - command byte
4	ZoomFactor
5	$?? - status
6	$?? - checksum

GetAllAverages()

Returns average value in display window for six images, three raw arrays and 3 bayer demultiplexed. Average values are 8 bits.

Message Byte	Contents
0	$02 - STX
1	$01 - LSB size
2	$00 - MSB size
3	$41 - command byte
4	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$08 - LSB size
2	$00 - MSB size
3	$41 - command byte
4	Array 1 Average
5	Array 2 Average
6	Array 3 Average
7	Bayer Red Average
8	Bayer Green Average
9	Bayer Blue Average
10	$?? - status
11	$?? - checksum

GetRemoteHeadConfiguration()

Used for remote head cameras only (RH1100 or RH1200). Returns configuration information for the three remote camera heads.

Byte 0 = HeadConfiguration;
- Bits 0,1;
  - Head 1 Configuration;
    - 0 = None
    - 1 = Monochrome
    - 2 = Color (Bayer)
    - 3 = Unused
- Bits 2,3;
  - Head 2 Configuration;
- Bits 4,5;
  - Head 3 Configuration;
- Bits 6,7; Unused

Message Byte	Contents
0	$02 - STX
1	$01 - LSB size
2	$00 - MSB size
3	$42 - command byte
4	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$03 - LSB size
2	$00 - MSB size
3	$42 - command byte
4	HeadConfiguration
5	$?? - status
6	$?? - checksum

SetBayerMux()

Used only with RH1100 Remote Head Area Scan cameras only. Specifies which head on the camera is to be used as the input source for the camera's Bayer Color Interpolator engine. Command Parameter:

Byte 0 = MuxConfiguration
0 = Array 1 as Input
1 = Array 2 as Input
2 = Array 3 as Input

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$43 - command byte
4	MuxConfiguration
5	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$02 - LSB size
2	$00 - MSB size
3	$43 - command byte
4	status
5	$?? - checksum

GetBayerMux()

Used only with RH1100 Remote Head Area Scan cameras. Queries the camera to get which head is presently assigned as input to the Bayer Color Interpolator engine. Command Parameter:

Byte 0 = MuxConfiguration
0 = Array 1 as Input
1 = Array 2 as Input
2 = Array 3 as Input

Message Byte	Contents
0	$02 - STX
1	$01 - LSB size
2	$00 - MSB size
3	$44 - command byte
4	$?? - checksum

Echo;

Message Byte	Contents
0	$02 - STX
1	$03 - LSB size
2	$00 - MSB size
3	$44 - command byte
4	MuxConfiguration
5	$?? - status
6	$?? - checksum